IMPORTING REQUIRED LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

READING THE DATASET

training data

```
df=pd.read_csv('/content/drive/MyDrive/Data set ML/churn-bigml-80.csv')
df
```

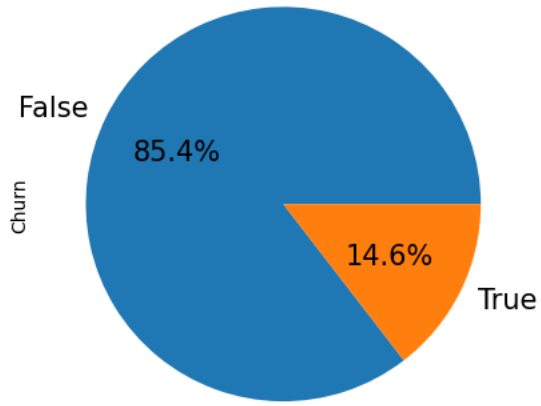
	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Tot e char
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	88	5
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12
...	...	...	...	...	...	...	...	...	...	...	...	
2661	SC	79	415	No	No	0	134.7	98	22.90	189.7	68	16
2662	AZ	192	415	No	Yes	36	156.2	77	26.55	215.5	126	18
2663	WV	68	415	No	No	0	231.1	57	39.29	153.4	55	13
2664	RI	28	510	No	No	0	180.8	109	30.74	288.8	58	24
2665	TN	74	415	No	Yes	25	234.4	113	39.85	265.9	82	22

2666 rows × 20 columns

CHURN COUNT GRAPH

```
df['Churn'].value_counts().plot(kind='pie',fontsize=15,autopct='%1.1f%%')
```

```
<Axes: ylabel='Churn'>
```




```
df.head()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.30
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	88	5.26
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61

```
df.tail()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Tot e char
2661	SC	79	415	No	No	0	134.7	98	22.90	189.7	68	16
2662	AZ	192	415	No	Yes	36	156.2	77	26.55	215.5	126	18
2663	WV	68	415	No	No	0	231.1	57	39.29	153.4	55	13
2664	RI	28	510	No	No	0	180.8	109	30.74	288.8	58	24
2665	TN	74	415	No	Yes	25	234.4	113	39.85	265.9	82	22

```
df.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
```

```
df.dtypes
```

```
State                    object
Account length            int64
Area code                 int64
International plan       object
Voice mail plan          object
Number vmail messages     int64
Total day minutes       float64
Total day calls           int64
Total day charge        float64
Total eve minutes       float64
Total eve calls           int64
Total eve charge        float64
Total night minutes     float64
Total night calls         int64
Total night charge      float64
Total intl minutes      float64
Total intl calls          int64
Total intl charge       float64
Customer service calls    int64
Churn                      bool
dtype: object
```

```
df.isna().sum()
```

```
State                   0
Account length          0
Area code               0
International plan      0
Voice mail plan         0
Number vmail messages   0
Total day minutes       0
Total day calls         0
Total day charge        0
Total eve minutes       0
```

```
Total eve calls           0
Total eve charge          0
Total night minutes       0
Total night calls         0
Total night charge        0
Total intl minutes        0
Total intl calls          0
Total intl charge         0
Customer service calls    0
Churn                     0
dtype: int64
```

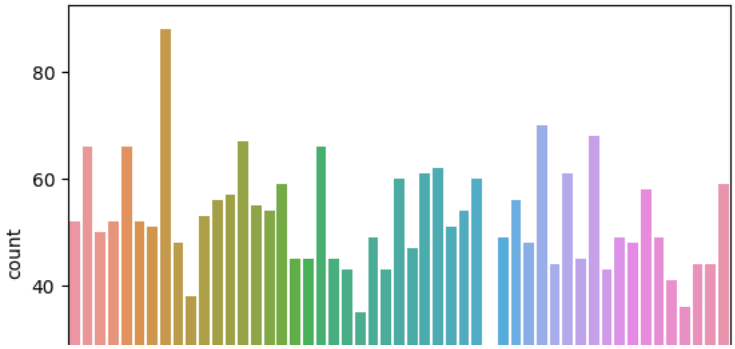
STATE COUNT GRAPH

```
df['State'].value_counts()
```

```
WV    88
MN    70
NY    68
VA    67
AL    66
OH    66
WY    66
OR    62
NV    61
WI    61
MD    60
UT    60
CO    59
CT    59
MI    58
VT    57
ID    56
NC    56
TX    55
FL    54
IN    54
MT    53
OK    52
MA    52
KS    52
MO    51
DE    51
NJ    50
SC    49
SD    49
ME    49
GA    49
RI    48
MS    48
WA    48
AR    47
IL    45
DC    45
AZ    45
NE    45
HI    44
NM    44
ND    44
AK    43
KY    43
NH    43
TN    41
IA    38
PA    36
LA    35
CA    24
Name: State, dtype: int64
```

```
sns.countplot(x='State',data=df)
```

```
<Axes: xlabel='State', ylabel='count'>
```



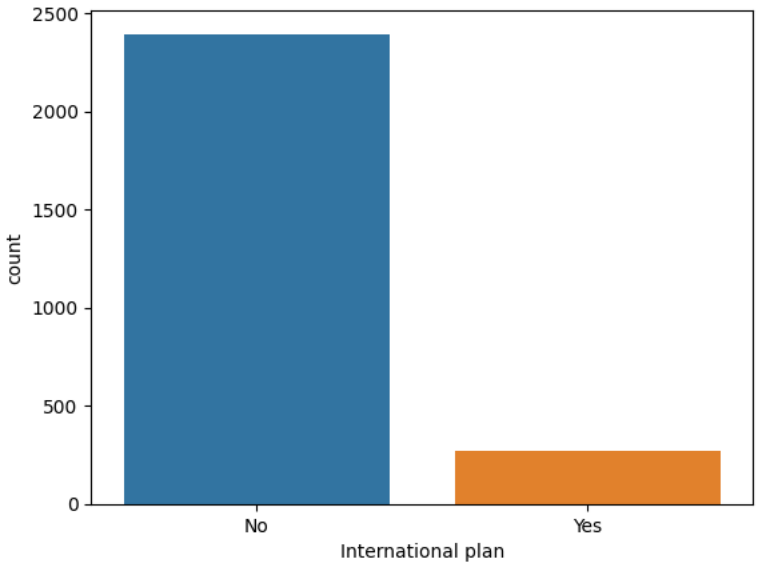

```
df['International plan'].value_counts()
```

```
No     2396
Yes     270
Name: International plan, dtype: int64
```

INTERNATIONAL PLAN COUNT GRAPH

```
sns.countplot(x='International plan',data=df)
```

```
<Axes: xlabel='International plan', ylabel='count'>
```




```
df['Voice mail plan'].value_counts()
```

```
No     1933
Yes     733
Name: Voice mail plan, dtype: int64
```

VOICE MAIL PLAN COUNT GRAPH

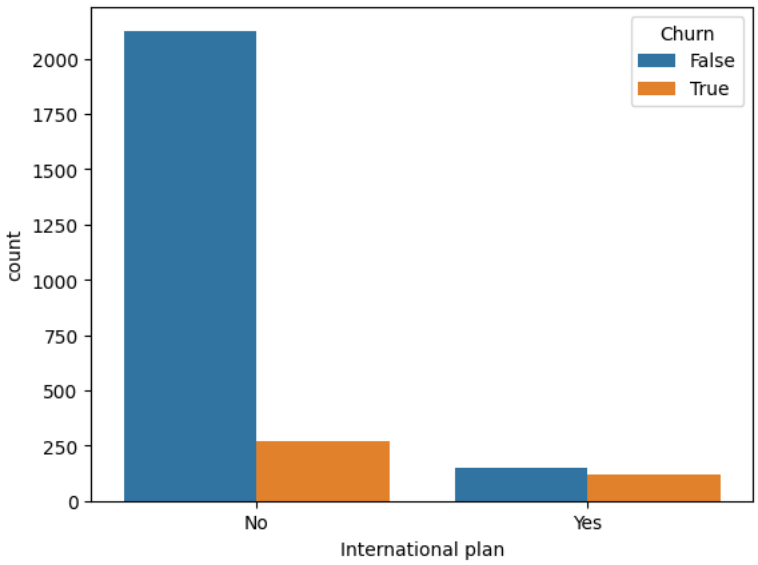
```
sns.countplot(x='Voice mail plan',data=df)
```

```
<Axes: xlabel='Voice mail plan', ylabel='count'>
```

International plan affecting churn graph

```
sns.countplot(x='International plan',hue='Churn',data=df)
```

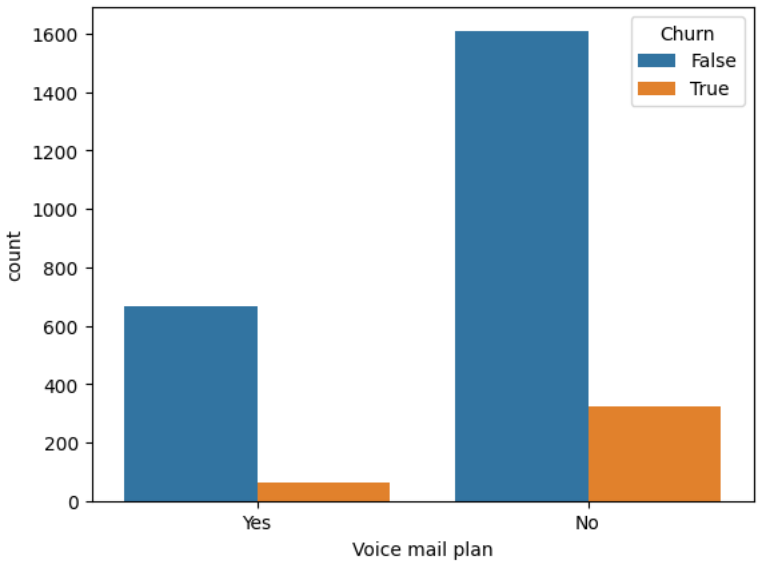
```
<Axes: xlabel='International plan', ylabel='count'>
```



voive mail affecting churn

```
sns.countplot(x='Voice mail plan',hue='Churn',data=df)
```

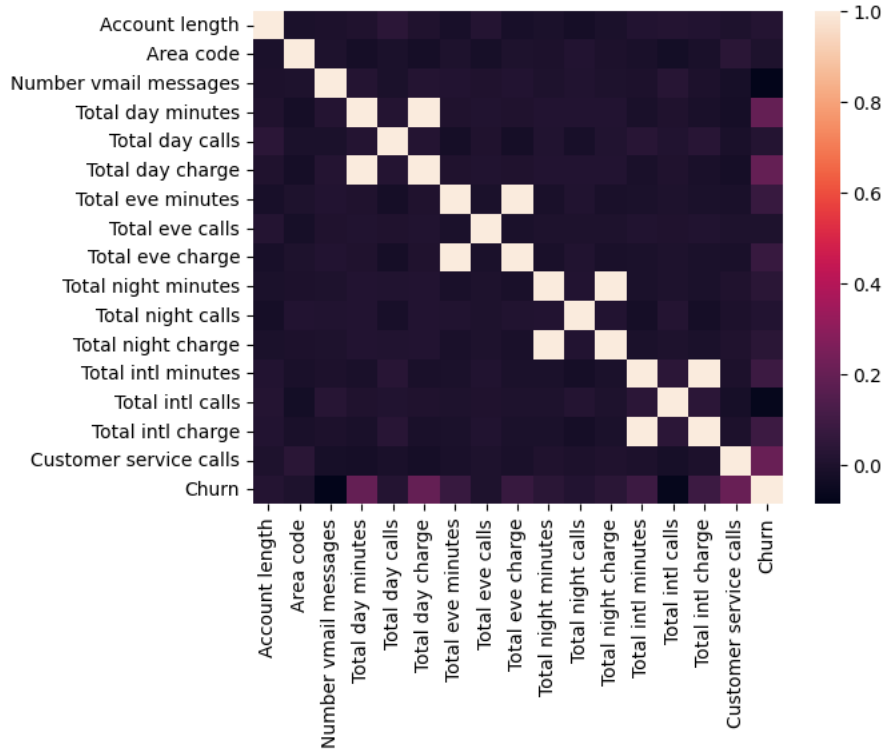
```
<Axes: xlabel='Voice mail plan', ylabel='count'>
```



HEAT MAP CORRELATION

```
sns.heatmap(df.corr())
```

```
<ipython-input-127-aa4f4450a243>:1: FutureWarning: The default value of numeric_only in DataFrame.corr
  sns.heatmap(df.corr())
<Axes: >
```




```
df1=pd.get_dummies(df[['State','International plan','Voice mail plan']],drop_first=True)
df1
```

	State_AL	State_AR	State_AZ	State_CA	State_CO	State_CT	State_DC	State_DE	State_FL	State_GA
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
2661	0	0	0	0	0	0	0	0	0	0
2662	0	0	1	0	0	0	0	0	0	0
2663	0	0	0	0	0	0	0	0	0	0
2664	0	0	0	0	0	0	0	0	0	0
2665	0	0	0	0	0	0	0	0	0	0

2666 rows × 52 columns

```
dfe=pd.concat([df,df1],axis=1)
dfe
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	...	State_
0	KS	128	415	No	Yes	25	265.1	110	45.07	197.4	...	
1	OH	107	415	No	Yes	26	161.6	123	27.47	195.5	...	
2	NJ	137	415	No	No	0	243.4	114	41.38	121.2	...	
3	OH	84	408	Yes	No	0	299.4	71	50.90	61.9	...	
4	OK	75	415	Yes	No	0	166.7	113	28.34	148.3	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2661	SC	79	415	No	No	0	134.7	98	22.90	189.7	...	
2662	AZ	192	415	No	Yes	36	156.2	77	26.55	215.5	...	
2663	WV	68	415	No	No	0	231.1	57	39.29	153.4	...	
2664	RI	28	510	No	No	0	180.8	109	30.74	288.8	...	

```
dfe.drop(['State','International plan','Voice mail plan'],axis=1,inplace=True)
```

```
dfe
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	...	State_TX	S
0	128	415	25	265.1	110	45.07	197.4	99	16.78	244.7	...	0	
1	107	415	26	161.6	123	27.47	195.5	103	16.62	254.4	...	0	
2	137	415	0	243.4	114	41.38	121.2	110	10.30	162.6	...	0	
3	84	408	0	299.4	71	50.90	61.9	88	5.26	196.9	...	0	
4	75	415	0	166.7	113	28.34	148.3	122	12.61	186.9	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2661	79	415	0	134.7	98	22.90	189.7	68	16.12	221.4	...	0	
2662	192	415	36	156.2	77	26.55	215.5	126	18.32	279.1	...	0	
2663	68	415	0	231.1	57	39.29	153.4	55	13.04	191.3	...	0	
2664	28	510	0	180.8	109	30.74	288.8	58	24.55	191.9	...	0	
2665	74	415	25	234.4	113	39.85	265.9	82	22.60	241.4	...	0	

2666 rows × 69 columns

```
x_train=dfe.drop(['Churn'],axis=1)
x_train
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	...	State_TX	S
0	128	415	25	265.1	110	45.07	197.4	99	16.78	244.7	...	0	
1	107	415	26	161.6	123	27.47	195.5	103	16.62	254.4	...	0	

```
y_train=df['Churn']
y_train
```

```
0       False
1       False
2       False
3       False
4       False
        ...
2661    False
2662    False
2663    False
2664    False
2665    False
Name: Churn, Length: 2666, dtype: bool
```

2666 rows × 68 columns

TESTING DATA

```
#testing data
df2=pd.read_csv('/content/drive/MyDrive/Data set ML/churn-bigml-20.csv')
df2
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Tota ev charg
0	LA	117	408	No	No	0	184.5	97	31.37	351.6	80	29.8
1	IN	65	415	No	No	0	129.1	137	21.95	228.5	83	19.4
2	NY	161	415	No	No	0	332.9	67	56.59	317.8	97	27.0
3	SC	111	415	No	No	0	110.4	103	18.77	137.3	102	11.6
4	HI	49	510	No	No	0	119.3	117	20.28	215.1	109	18.2
...	...	...	...	...	...	...	...	...	...	...	...	
662	WI	114	415	No	Yes	26	137.1	88	23.31	155.7	125	13.2
663	AL	106	408	No	Yes	29	83.6	131	14.21	203.9	131	17.3
664	VT	60	415	No	No	0	193.9	118	32.96	85.0	110	7.2
665	WV	159	415	No	No	0	169.8	114	28.87	197.7	105	16.8
666	CT	184	510	Yes	No	0	213.8	105	36.35	159.6	84	13.5

667 rows × 20 columns

```
df2.head()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge
0	LA	117	408	No	No	0	184.5	97	31.37	351.6	80	29.89
1	IN	65	415	No	No	0	129.1	137	21.95	228.5	83	19.42
2	NY	161	415	No	No	0	332.9	67	56.59	317.8	97	27.01
3	SC	111	415	No	No	0	110.4	103	18.77	137.3	102	11.67
4	HI	49	510	No	No	0	119.3	117	20.28	215.1	109	18.28

```
df2.tail()
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Tota ev charg
662	WI	114	415	No	Yes	26	137.1	88	23.31	155.7	125	13.2
663	AL	106	408	No	Yes	29	83.6	131	14.21	203.9	131	17.3
664	VT	60	415	No	No	0	193.9	118	32.96	85.0	110	7.2
665	WV	159	415	No	No	0	169.8	114	28.87	197.7	105	16.8
666	CT	184	510	Yes	No	0	213.8	105	36.35	159.6	84	13.5

```
df2.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl minutes',
       'Total intl calls', 'Total intl charge', 'Customer service calls',
       'Churn'],
      dtype='object')
```

```
df2.dtypes
```

```
State                    object
Account length            int64
Area code                 int64
International plan       object
Voice mail plan          object
Number vmail messages     int64
Total day minutes       float64
Total day calls           int64
Total day charge        float64
Total eve minutes       float64
Total eve calls           int64
Total eve charge        float64
Total night minutes     float64
Total night calls         int64
Total night charge      float64
Total intl minutes      float64
Total intl calls          int64
Total intl charge       float64
Customer service calls    int64
Churn                      bool
dtype: object
```

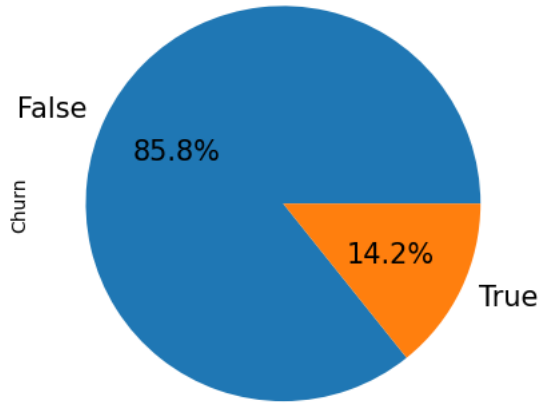
```
df2.isna().sum()
```

```
State                   0
Account length          0
Area code               0
International plan      0
Voice mail plan         0
Number vmail messages   0
Total day minutes       0
Total day calls         0
Total day charge        0
Total eve minutes       0
Total eve calls         0
Total eve charge        0
Total night minutes     0
Total night calls       0
Total night charge      0
Total intl minutes      0
Total intl calls        0
Total intl charge       0
Customer service calls  0
Churn                   0
dtype: int64
```

CHURN COUNT GRAPH

```
df2['Churn'].value_counts().plot(kind='pie',fontsize=15,autopct='%1.1f%%')
```

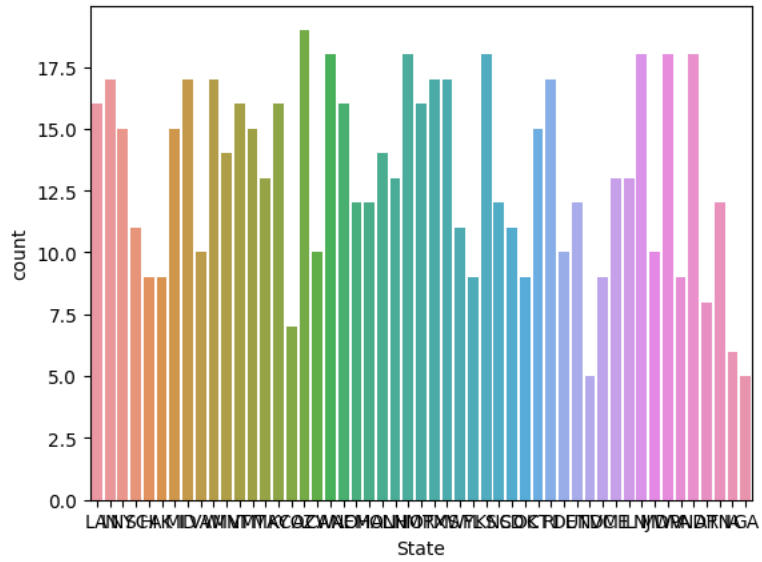
```
<Axes: ylabel='Churn'>
```



STATE COUNT GRAPH

```
sns.countplot(x='State',data=df2)
```

```
<Axes: xlabel='State', ylabel='count'>
```



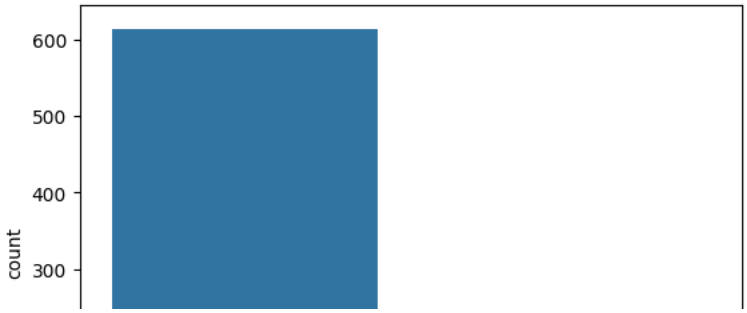

```
df2['International plan'].value_counts()
```

```
No     614
Yes     53
Name: International plan, dtype: int64
```

INTERNATIONAL PLAN COUNT GRAPH

```
sns.countplot(x='International plan',data=df2)
```

```
<Axes: xlabel='International plan', ylabel='count'>
```



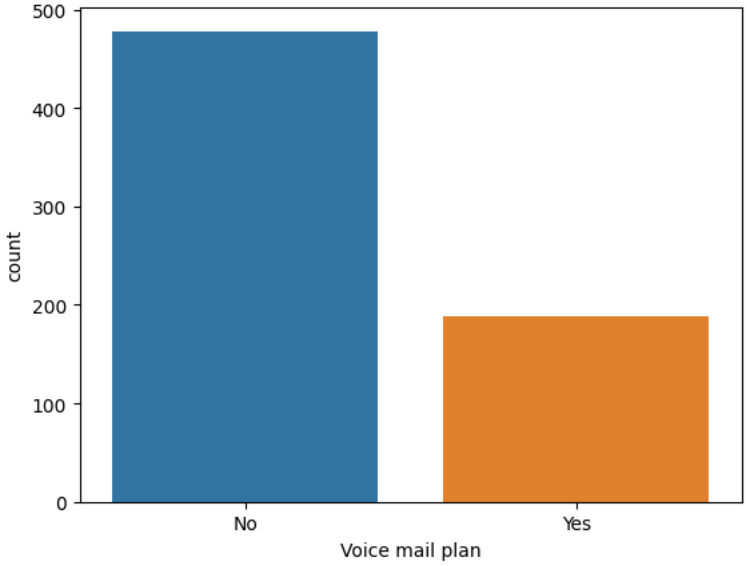

```
df2['Voice mail plan'].value_counts()
```

```
No     478
Yes    189
Name: Voice mail plan, dtype: int64
```

VOICE MAIL PLAN COUNT GRAPH

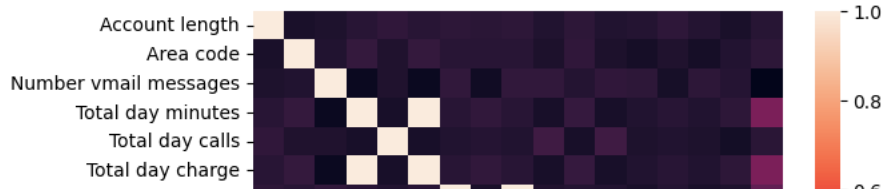
```
sns.countplot(x='Voice mail plan',data=df2)
```

```
<Axes: xlabel='Voice mail plan', ylabel='count'>
```




```
sns.heatmap(df2.corr())
```

```
<ipython-input-145-28890f610699>:1: FutureWarning: The default value of numeric_only in DataFrame.corr
  sns.heatmap(df2.corr())
<Axes: >
```




```
df3=pd.get_dummies(df2[['State','International plan','Voice mail plan']],drop_first=True)
df3
```

	State_AL	State_AR	State_AZ	State_CA	State_CO	State_CT	State_DC	State_DE	State_FL	State_GA
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
662	0	0	0	0	0	0	0	0	0	0
663	1	0	0	0	0	0	0	0	0	0
664	0	0	0	0	0	0	0	0	0	0
665	0	0	0	0	0	0	0	0	0	0
666	0	0	0	0	0	1	0	0	0	0

667 rows × 52 columns

```
dfg=pd.concat([df2,df3],axis=1)
dfg
```

	State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	...	State_1
0	LA	117	408	No	No	0	184.5	97	31.37	351.6	...	
1	IN	65	415	No	No	0	129.1	137	21.95	228.5	...	
2	NY	161	415	No	No	0	332.9	67	56.59	317.8	...	
3	SC	111	415	No	No	0	110.4	103	18.77	137.3	...	
4	HI	49	510	No	No	0	119.3	117	20.28	215.1	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
662	WI	114	415	No	Yes	26	137.1	88	23.31	155.7	...	
663	AL	106	408	No	Yes	29	83.6	131	14.21	203.9	...	
664	VT	60	415	No	No	0	193.9	118	32.96	85.0	...	
665	WV	159	415	No	No	0	169.8	114	28.87	197.7	...	
666	CT	184	510	Yes	No	0	213.8	105	36.35	159.6	...	

667 rows × 72 columns

```
dfg.drop(['State','International plan','Voice mail plan'],axis=1,inplace=True)
dfg
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	...	State_TX	St
0	117	408	0	184.5	97	31.37	351.6	80	29.89	215.8	...	0	
1	65	415	0	129.1	137	21.95	228.5	83	19.42	208.8	...	0	
2	161	415	0	332.9	67	56.59	317.8	97	27.01	160.6	...	0	
3	111	415	0	110.4	103	18.77	137.3	102	11.67	189.6	...	0	
4	49	510	0	119.3	117	20.28	215.1	109	18.28	178.7	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
662	114	415	26	137.1	88	23.31	155.7	125	13.23	247.6	...	0	
663	106	408	29	83.6	131	14.21	203.9	131	17.33	229.5	...	0	
664	60	415	0	193.9	118	32.96	85.0	110	7.23	210.1	...	0	
665	159	415	0	169.8	114	28.87	197.7	105	16.80	193.7	...	0	
666	184	510	0	213.8	105	36.35	159.6	84	13.57	139.2	...	0	

667 rows × 69 columns

```
x_test=dfg.drop(['Churn'],axis=1)
x_test.head()
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	...	State_TX	Stat
0	117	408	0	184.5	97	31.37	351.6	80	29.89	215.8	...	0	
1	65	415	0	129.1	137	21.95	228.5	83	19.42	208.8	...	0	
2	161	415	0	332.9	67	56.59	317.8	97	27.01	160.6	...	0	
3	111	415	0	110.4	103	18.77	137.3	102	11.67	189.6	...	0	
4	49	510	0	119.3	117	20.28	215.1	109	18.28	178.7	...	0	

5 rows × 68 columns

```
y_test=dfg["Churn"]
y_test
```

```
0      False
1       True
2       True
3      False
4      False
       ...
662    False
663    False
664    False
665    False
666    False
Name: Churn, Length: 667, dtype: bool
```

```
# x_train=x,y_train=y
```

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(x_train)
x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)
x_train
```

```
array([[0.52479339, 0.06862745, 0.5       , ..., 0.        , 0.        ,
        1.        ],
       [0.43801653, 0.06862745, 0.52      , ..., 0.        , 0.        ,
```

```
             1.        ],
            [0.56198347, 0.06862745, 0.        , ..., 0.        , 0.        ,
             0.        ],
            ...,
            [0.2768595 , 0.06862745, 0.        , ..., 0.        , 0.        ,
             0.        ],
            [0.11157025, 1.        , 0.        , ..., 0.        , 0.        ,
             0.        ],
            [0.30165289, 0.06862745, 0.5       , ..., 0.        , 0.        ,
             1.        ]])
```

```
x_test=scaler.fit_transform(x_test)
x_test
```

```
    array([[0.5021645 , 0.        , 0.        , ..., 0.        , 0.        ,
            0.        ],
           [0.27705628, 0.06862745, 0.        , ..., 0.        , 0.        ,
            0.        ],
           [0.69264069, 0.06862745, 0.        , ..., 0.        , 0.        ,
            0.        ],
           ...,
           [0.25541126, 0.06862745, 0.        , ..., 0.        , 0.        ,
            0.        ],
           [0.68398268, 0.06862745, 0.        , ..., 0.        , 0.        ,
            0.        ],
           [0.79220779, 1.        , 0.        , ..., 0.        , 1.        ,
            0.        ]])
```

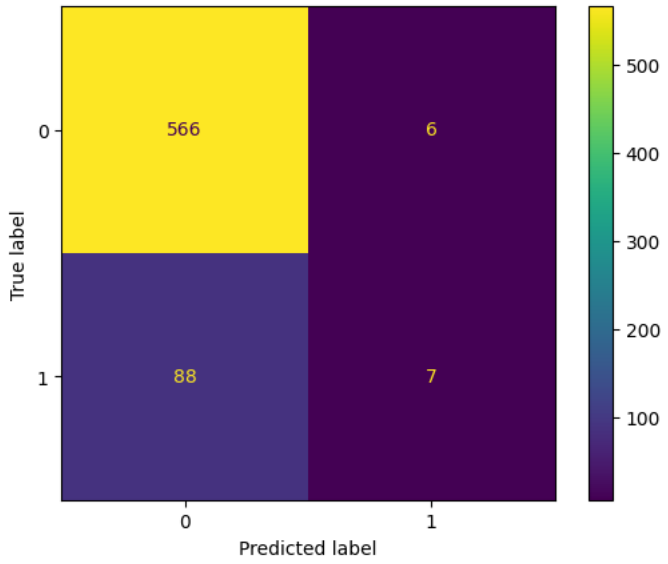
```
#model

from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
K_model=KNeighborsClassifier(n_neighbors=5)
nb_model=MultinomialNB()
sv_model=SVC()
lst_model=[K_model,nb_model,sv_model]
```

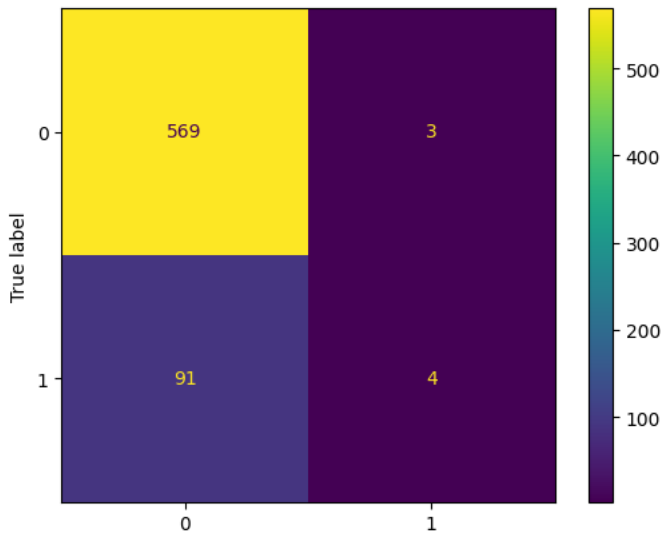
```
y_test
```

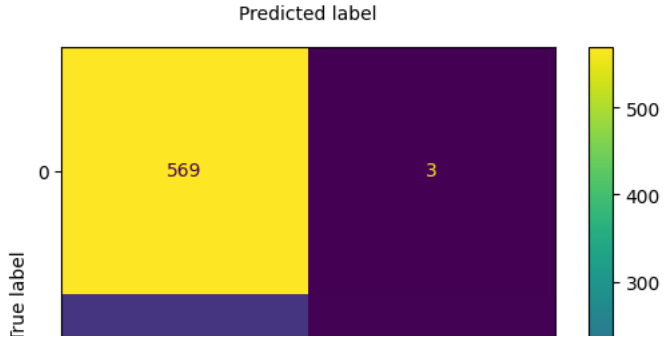
```
    0      False
    1       True
    2       True
    3      False
    4      False
           ...
    662    False
    663    False
    664    False
    665    False
    666    False
    Name: Churn, Length: 667, dtype: bool
```

```
from sklearn.metrics import ConfusionMatrixDisplay,confusion_matrix,classification_report,accuracy_score
cm=['0','1']
for i in lst_model:
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print(i)
  result=confusion_matrix(y_test,y_pred)
  print(accuracy_score(y_test,y_pred))
  print(classification_report(y_test,y_pred))
  cmd=ConfusionMatrixDisplay(result,display_labels=cm)
  cmd.plot()
  print('*'*100)
```

```
KNeighborsClassifier()
0.8590704647676162
              precision    recall  f1-score   support

       False       0.87      0.99      0.92       572
        True       0.54      0.07      0.13        95

    accuracy                           0.86       667
   macro avg       0.70      0.53      0.53       667
weighted avg       0.82      0.86      0.81       667

**********************************************************************************************************
MultinomialNB()
0.8590704647676162
              precision    recall  f1-score   support

       False       0.86      0.99      0.92       572
        True       0.57      0.04      0.08        95

    accuracy                           0.86       667
   macro avg       0.72      0.52      0.50       667
weighted avg       0.82      0.86      0.80       667

**********************************************************************************************************
SVC()
0.863568215892054
              precision    recall  f1-score   support

       False       0.87      0.99      0.93       572
        True       0.70      0.07      0.13        95

    accuracy                           0.86       667
   macro avg       0.78      0.53      0.53       667
weighted avg       0.84      0.86      0.81       667

**********************************************************************************************************
```







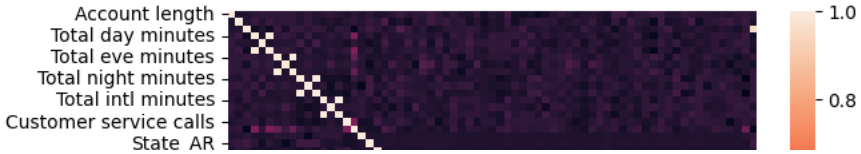

```
dfg.corr()
```

	Account length	Area code	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge
Account length	1.000000	-0.026327	-0.011993	0.017833	0.035703	0.017839	0.027043	0.021237	0.027051
Area code	-0.026327	1.000000	-0.006907	0.051507	-0.008972	0.051492	0.017160	0.017783	0.017182
Number vmail messages	-0.011993	-0.006907	1.000000	-0.069172	-0.009952	-0.069187	0.040865	-0.051951	0.040876
Total day minutes	0.017833	0.051507	-0.069172	1.000000	-0.032306	1.000000	0.017987	0.043219	0.017945
Total day calls	0.035703	-0.008972	-0.009952	-0.032306	1.000000	-0.032319	-0.004688	0.005851	-0.004664
...	...	...	...	...	...	...	...	...	...
State_WI	0.012058	0.039947	0.002087	0.029805	-0.042908	0.029802	-0.022698	0.005976	-0.022683
State_WV	-0.010919	0.055215	-0.011473	0.007445	-0.005387	0.007446	-0.009683	-0.077491	-0.009760
State_WY	0.016089	0.010790	-0.045867	0.001458	0.041986	0.001449	-0.060566	-0.005748	-0.060597
International plan_Yes	0.029549	0.053461	0.023145	0.050915	0.040889	0.050903	-0.011273	-0.012082	-0.011274
Voice mail plan_Yes	0.003918	-0.032253	0.956168	-0.060599	-0.025318	-0.060612	0.030493	-0.048341	0.030501

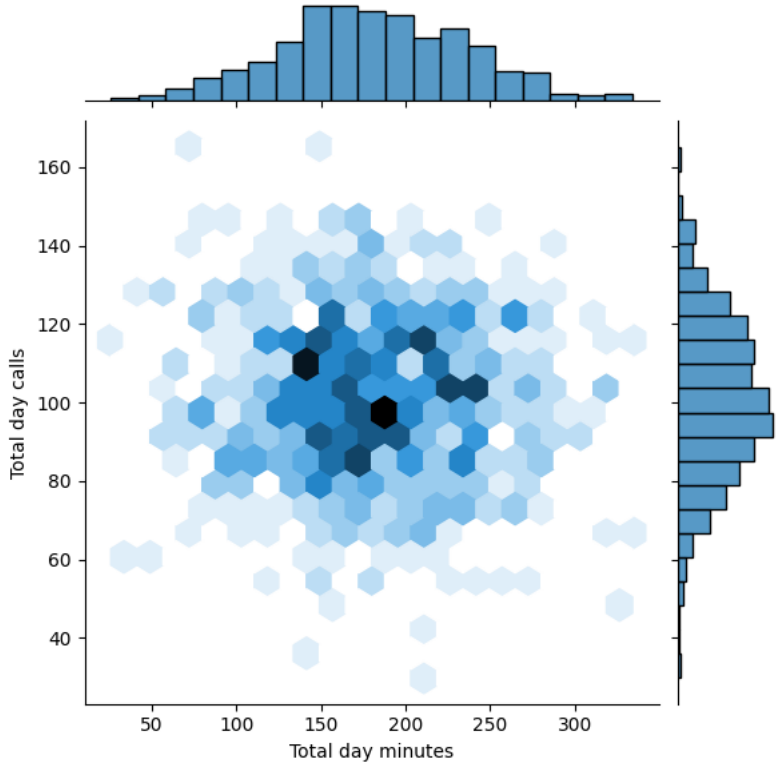
69 rows × 69 columns

```
#heatmap
sns.heatmap(dfg.corr())
```

```
<Axes: >
```




```
#jointplot

sns.jointplot(x='Total day minutes',y='Total day calls',data=dfg,kind='hex')

<seaborn.axisgrid.JointGrid at 0x7e66e5c55db0>
```



✓ 0s completed at 11:58 AM